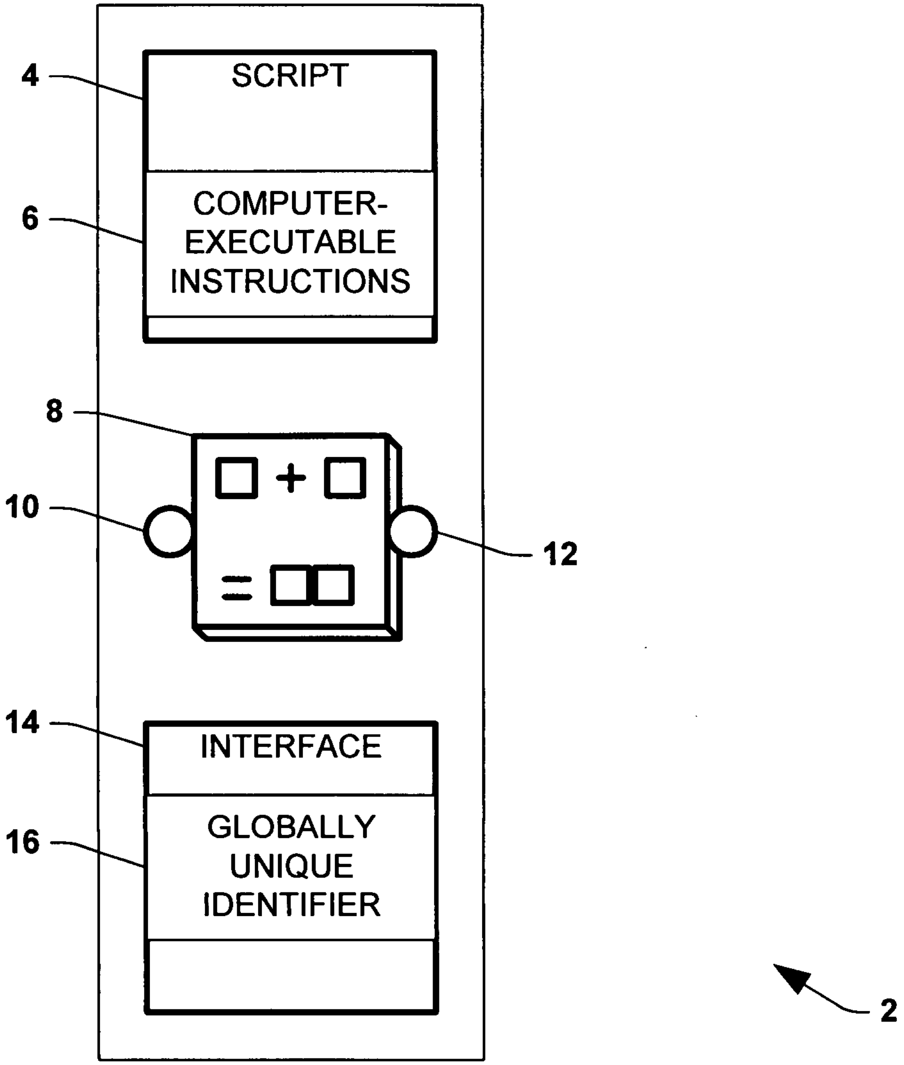

4
SCRIPT
6
COMPUTER-
EXECUTABLE
INSTRUCTIONS
8
□ + □
= □□
10
12
14
INTERFACE
16
GLOBALLY
UNIQUE
IDENTIFIER
2

FIG. 1

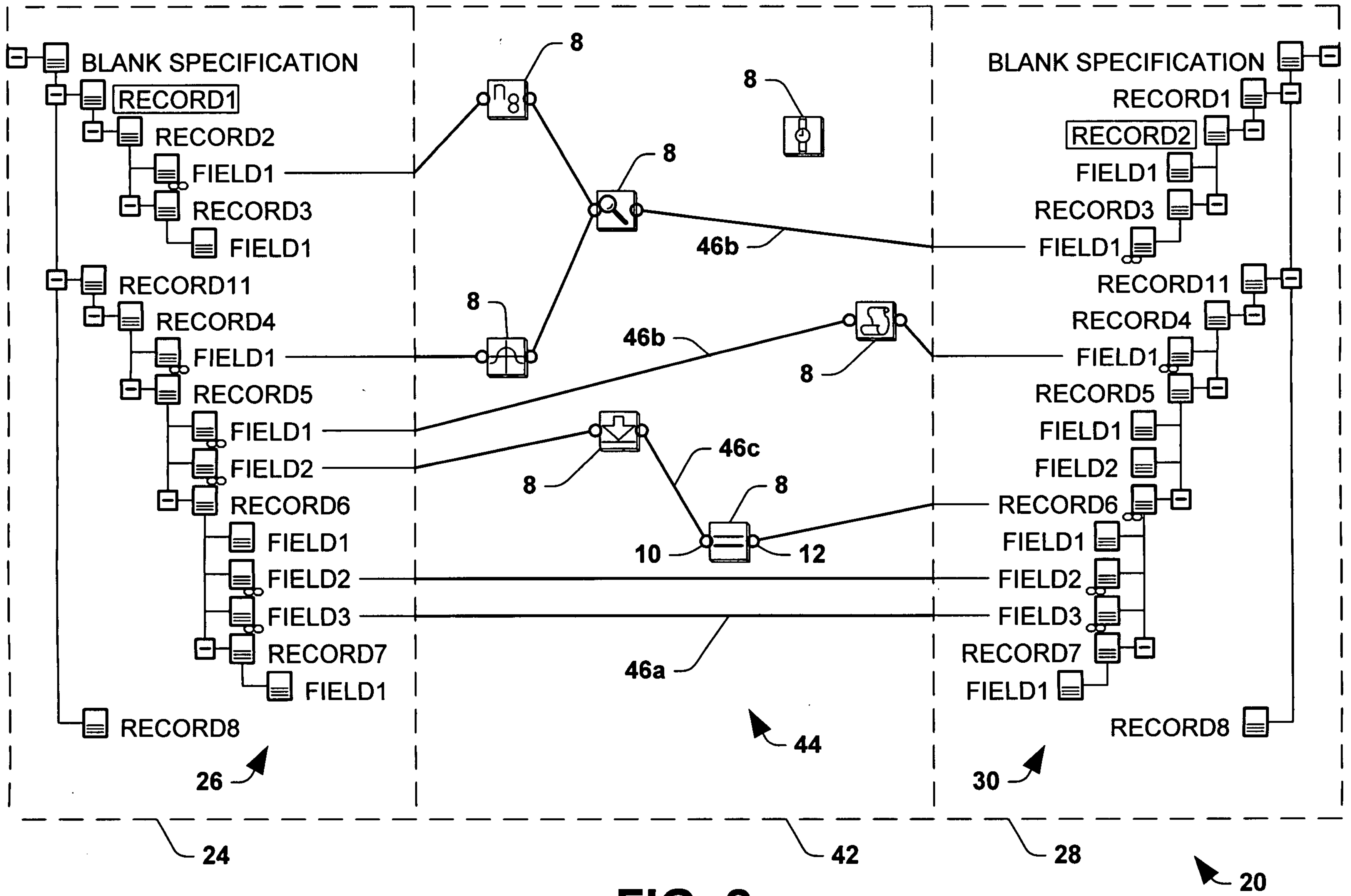

BLANK SPECIFICATION
RECORD1
RECORD2
FIELD1
RECORD3
FIELD1
RECORD11
RECORD4
FIELD1
RECORD5
FIELD1
FIELD2
RECORD6
FIELD1
FIELD2
FIELD3
RECORD7
FIELD1
RECORD8
26
24
8
8
8
8
8
8
8
8
46b
46b
46c
46a
10
12
44
42
BLANK SPECIFICATION
RECORD1
RECORD2
FIELD1
RECORD3
FIELD1
RECORD11
RECORD4
FIELD1
RECORD5
FIELD1
FIELD2
RECORD6
FIELD1
FIELD2
FIELD3
RECORD7
FIELD1
RECORD8
30
28
20

FIG. 2

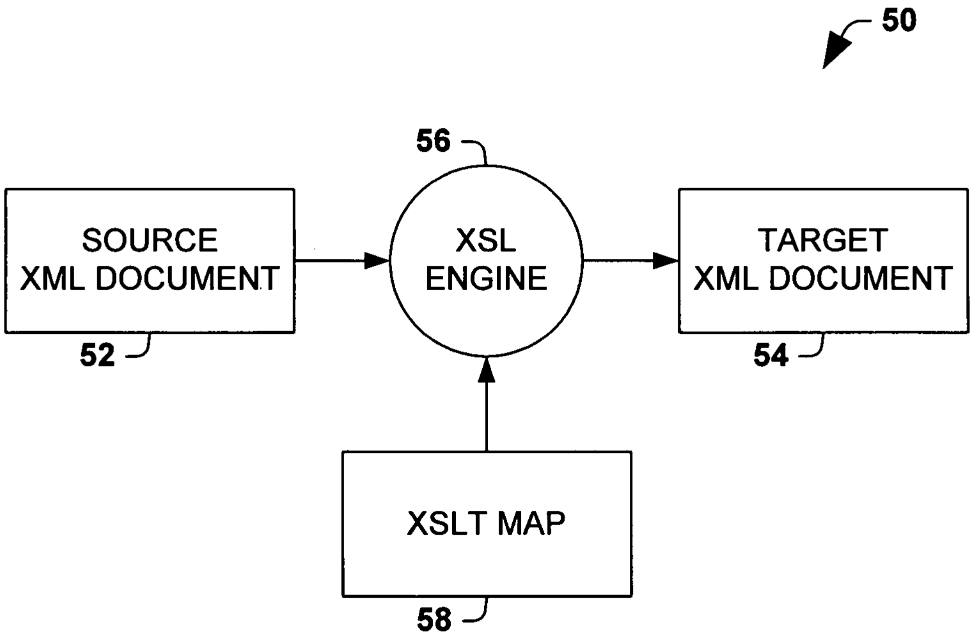
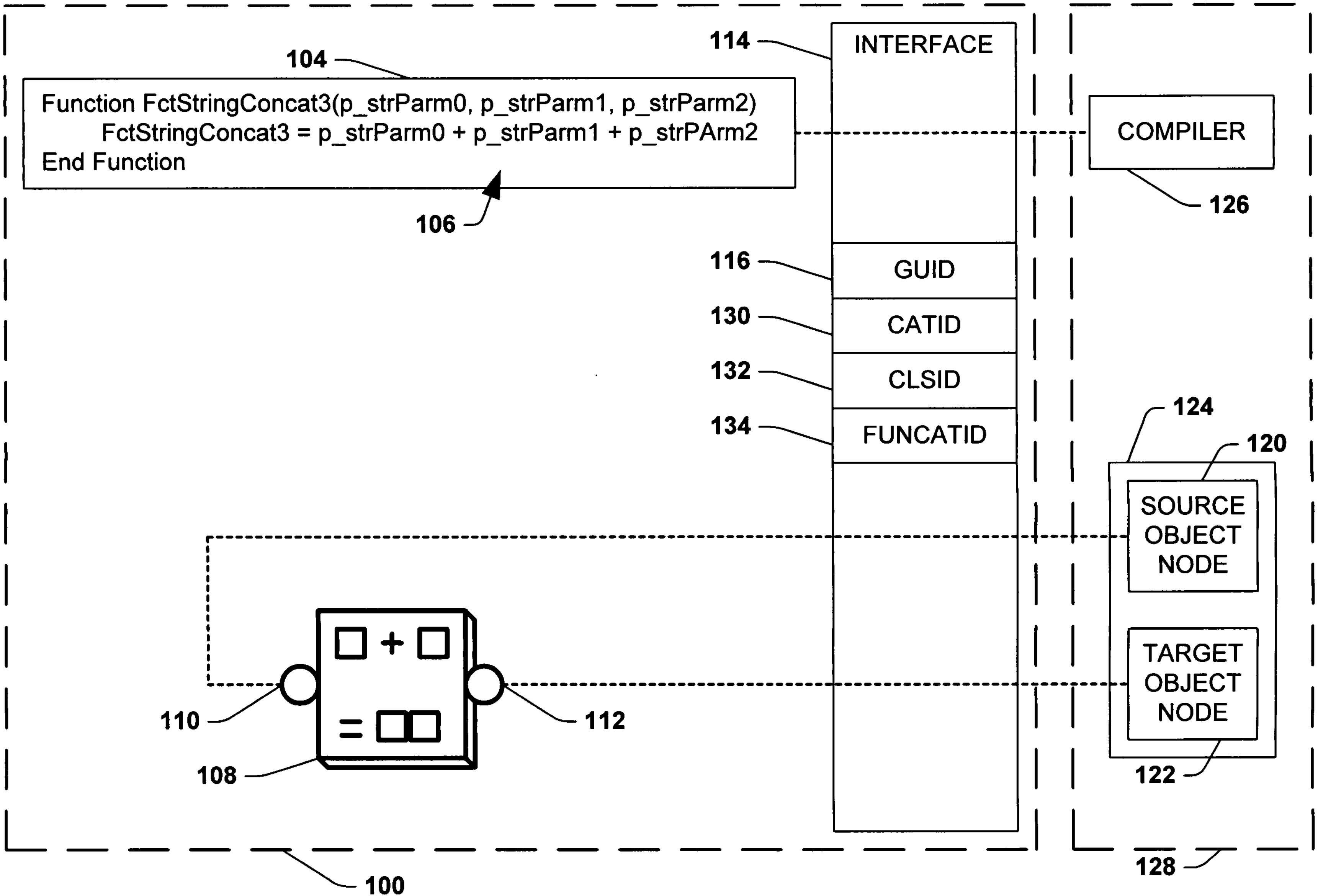

50
56
SOURCE XML DOCUMENT
52
XSL ENGINE
TARGET XML DOCUMENT
54
XSLT MAP
58

FIG. 3



```
Function FctStringConcat3(p_strParm0, p_strParm1, p_strParm2)
      FctStringConcat3 = p_strParm0 + p_strParm1 + p_strPArm2
End Function
```

FIG. 4A

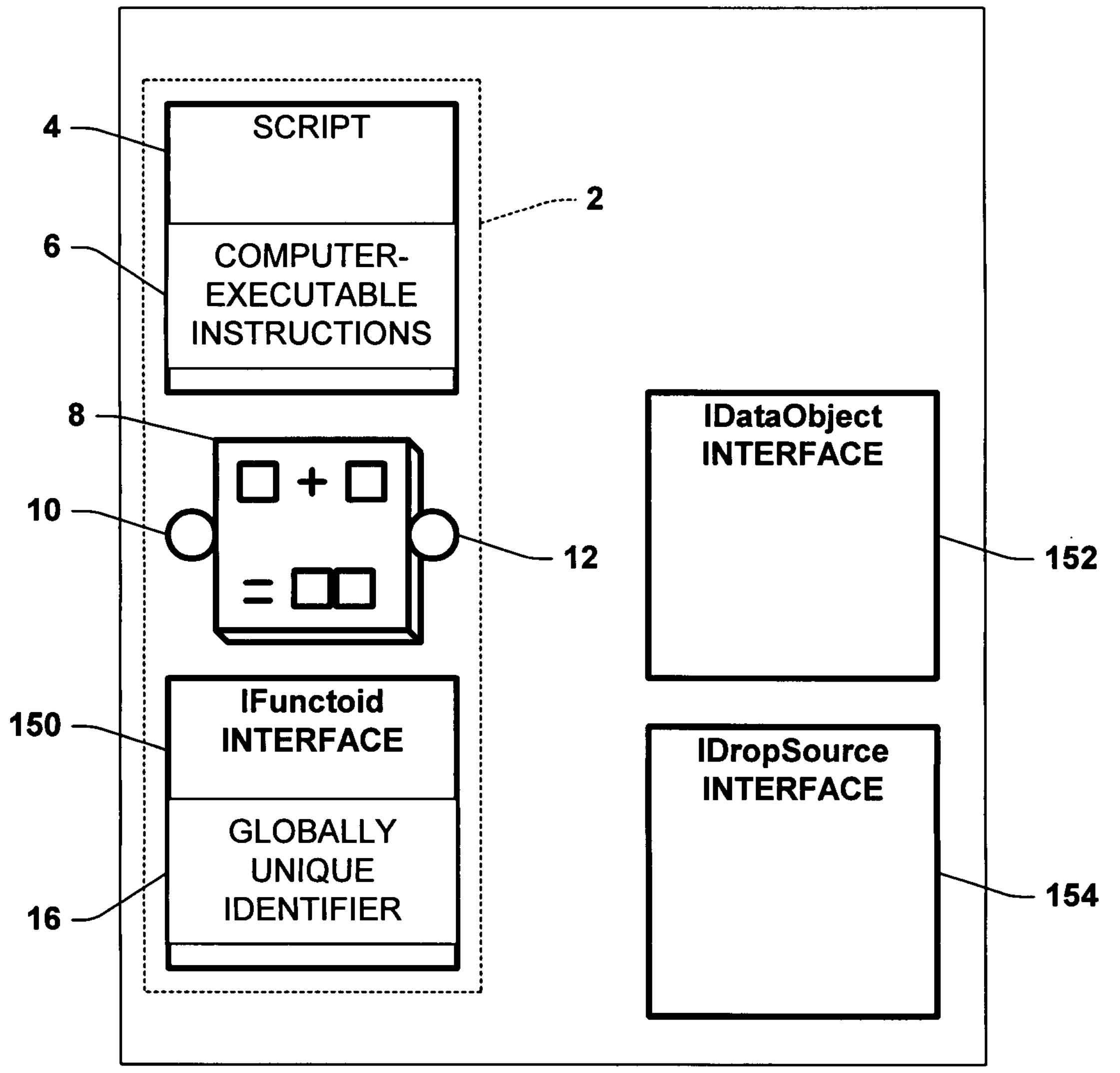

4
SCRIPT
6
COMPUTER-EXECUTABLE INSTRUCTIONS
2
8
10
12
=
+
150
IFunctoid INTERFACE
16
GLOBALLY UNIQUE IDENTIFIER
IDataObject INTERFACE
152
IDropSource INTERFACE
154
156

FIG. 4B

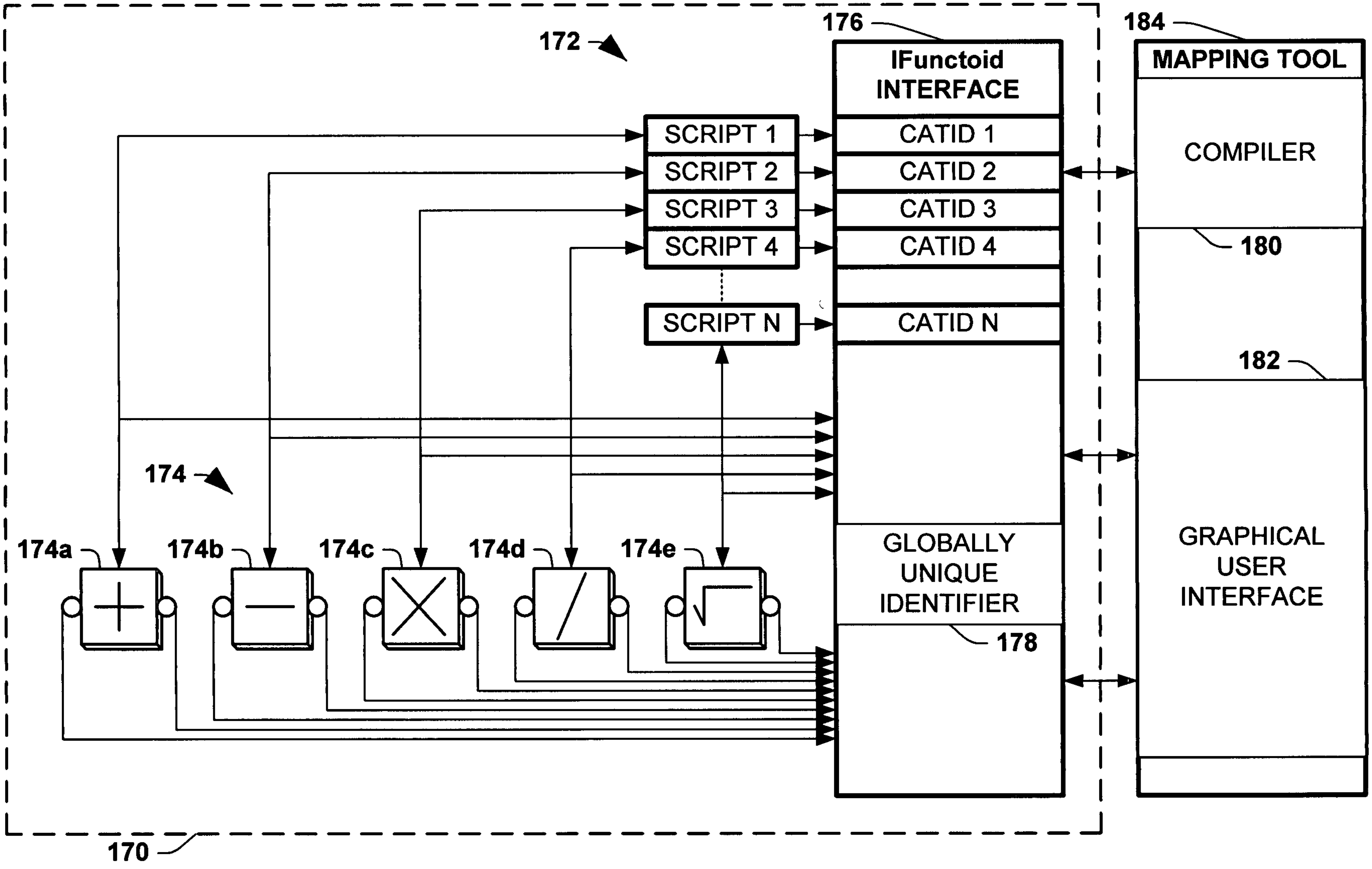

172
176
IFunctoid INTERFACE
184
MAPPING TOOL
SCRIPT 1
SCRIPT 2
SCRIPT 3
SCRIPT 4
SCRIPT N
CATID 1
CATID 2
CATID 3
CATID 4
CATID N
COMPILER
180
182
174
174a
174b
174c
174d
174e
GLOBALLY UNIQUE IDENTIFIER
178
GRAPHICAL USER INTERFACE
170

FIG. 4C

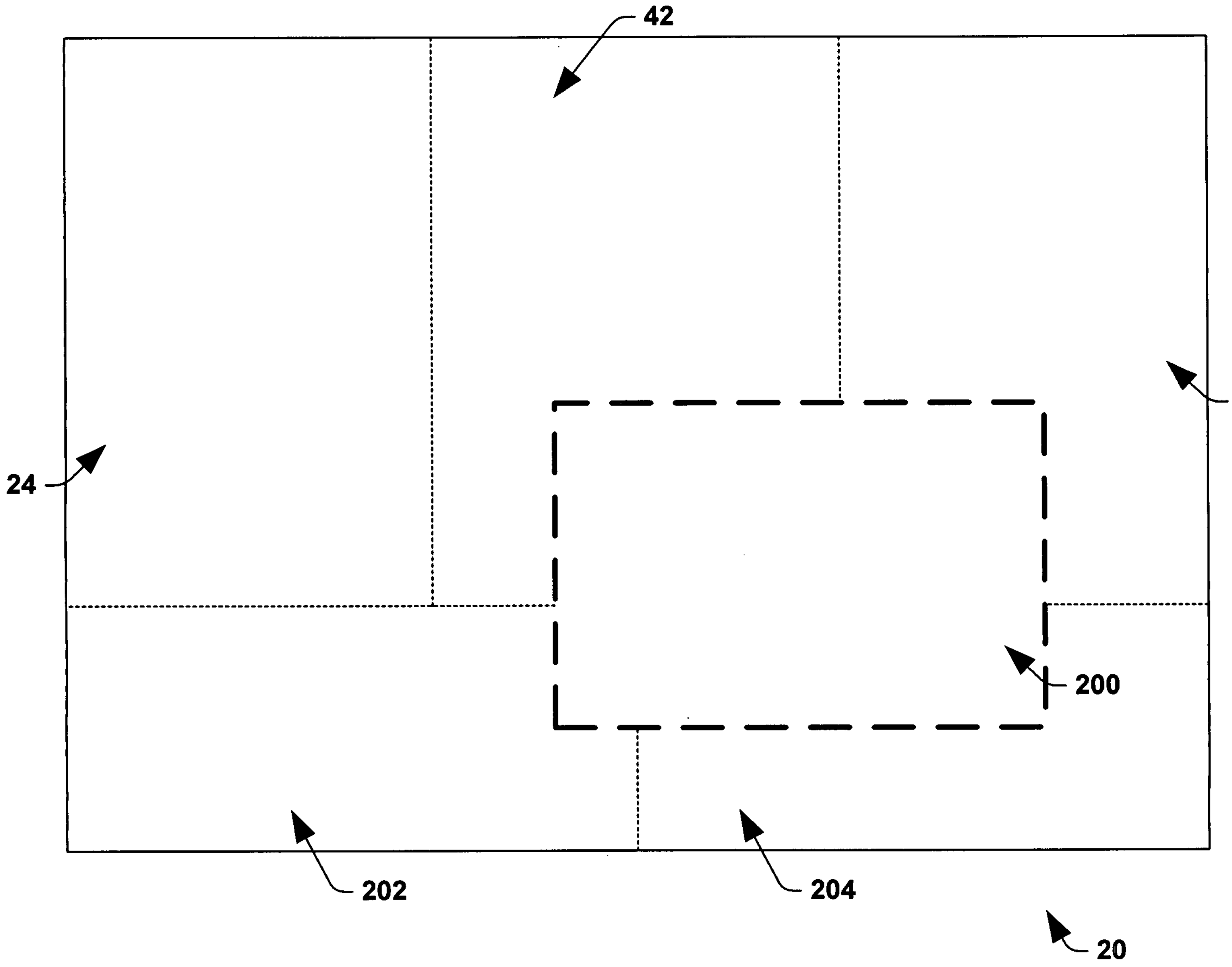


FIG. 5A

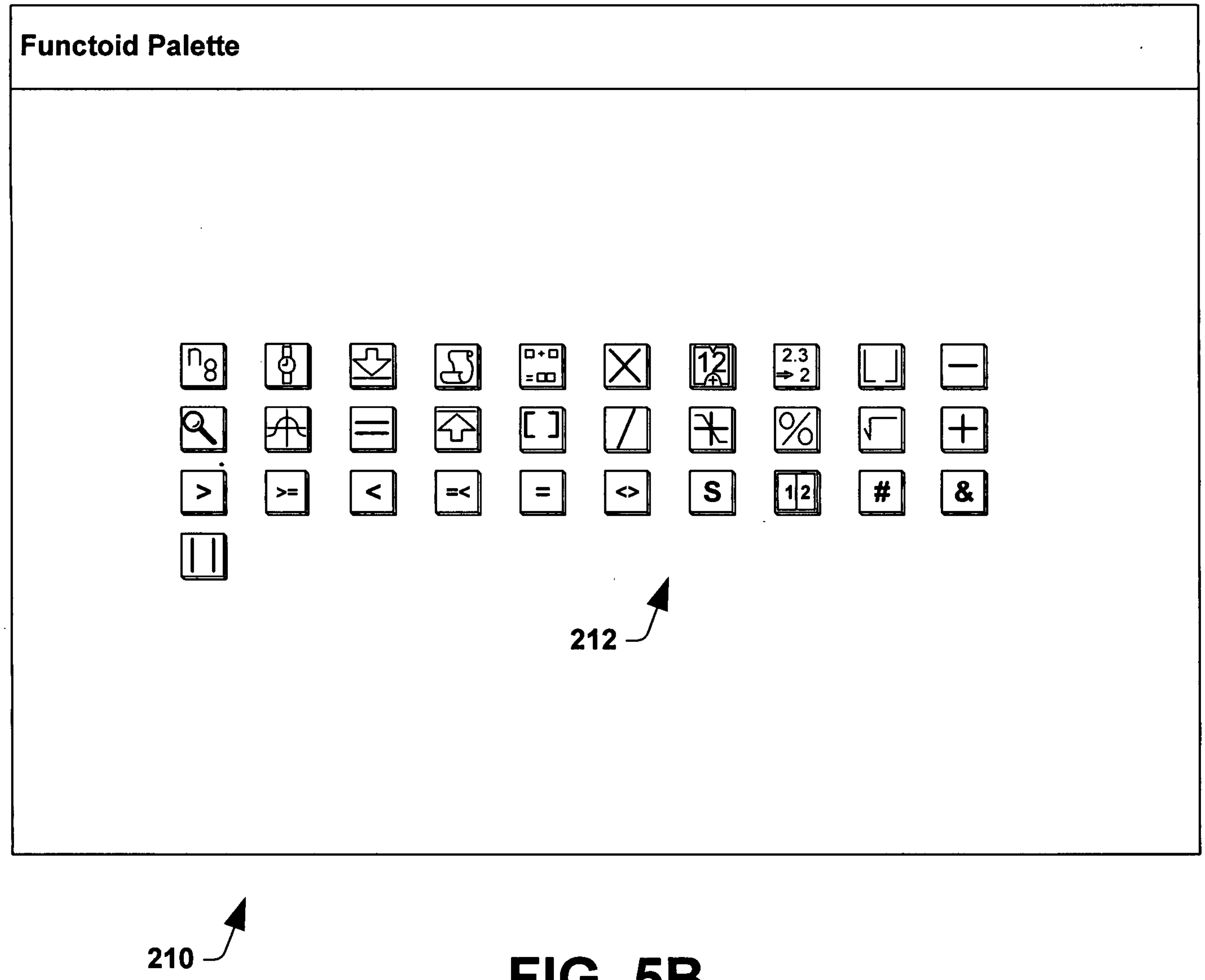


FIG. 5B

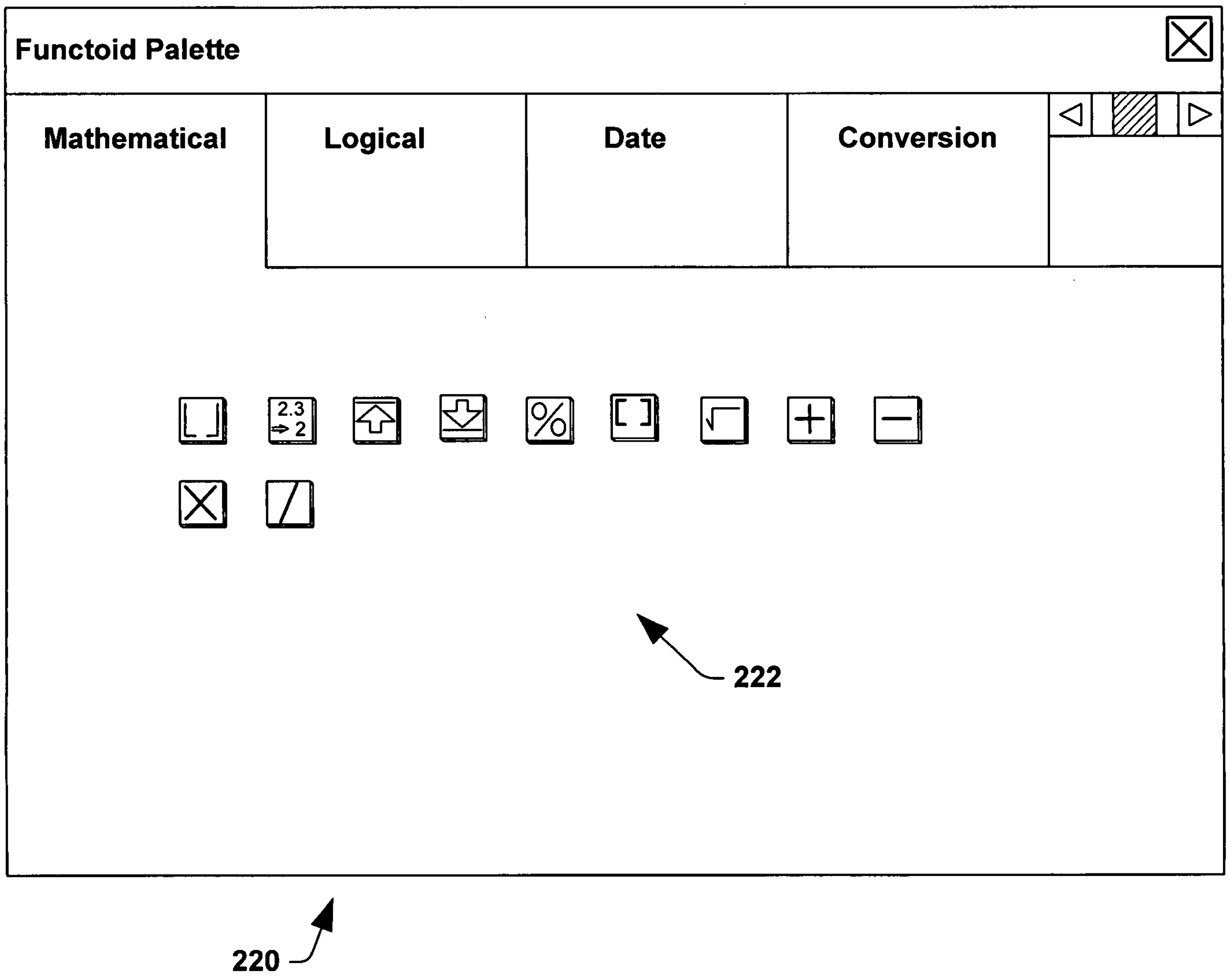

Functoid Palette
Mathematical
Logical
Date
Conversion
2.3
2
%
222
220

FIG. 5C

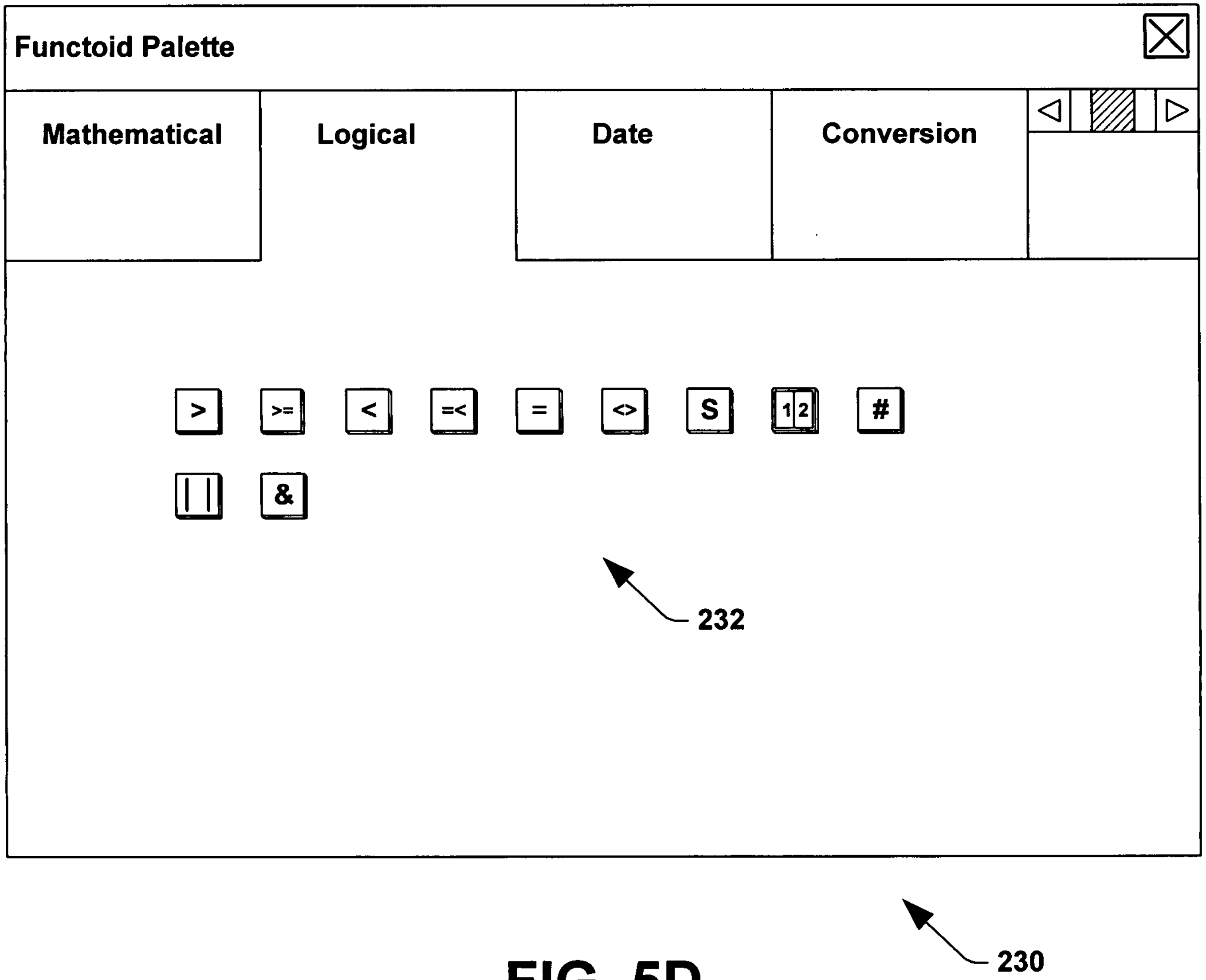

Functoid Palette
Mathematical
Logical
Date
Conversion
>
>=
<
=<
=
<>
S
1 2
#
||
&
232
230

FIG. 5D

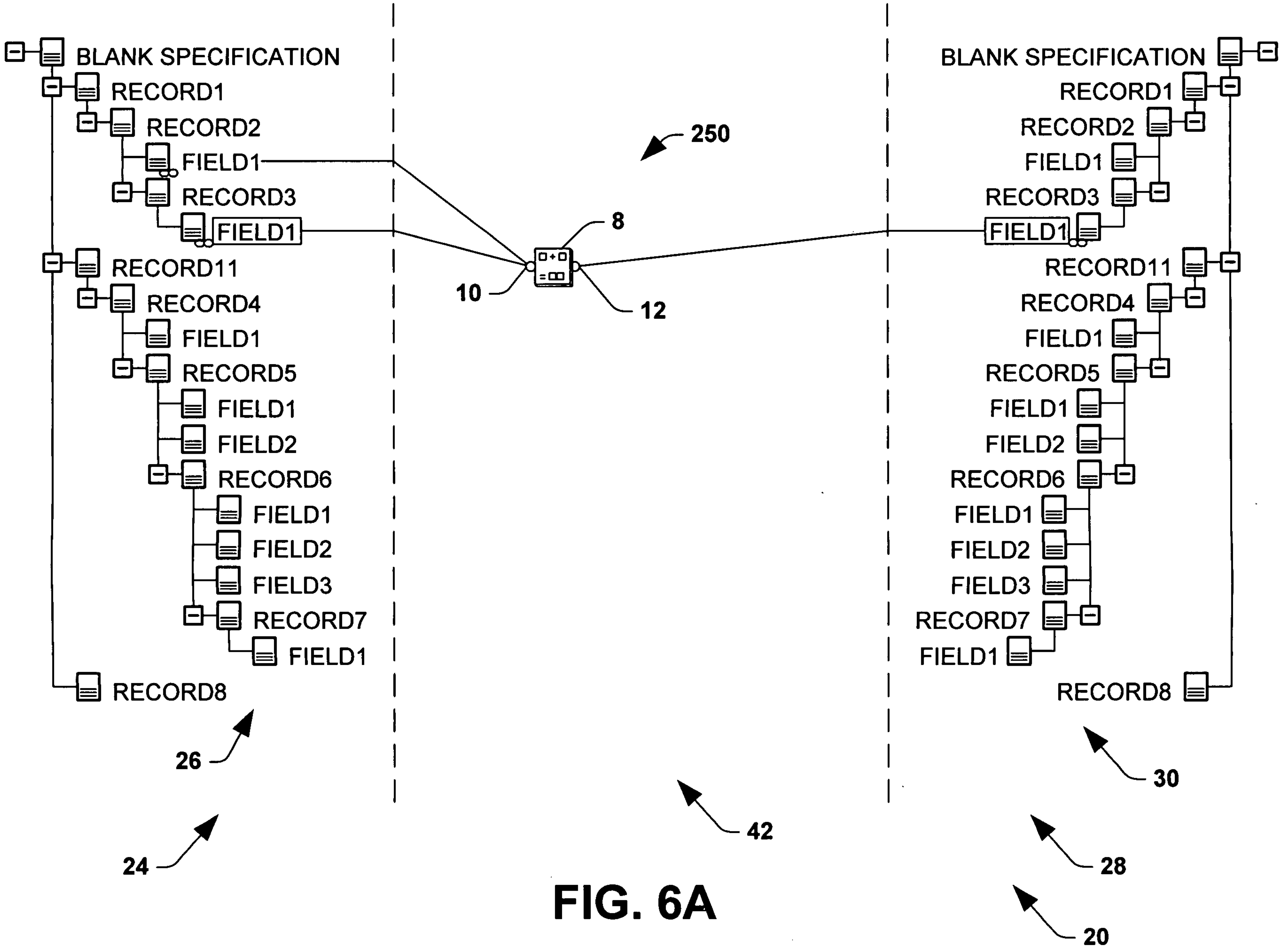

BLANK SPECIFICATION
RECORD1
RECORD2
FIELD1
RECORD3
FIELD1
RECORD11
RECORD4
FIELD1
RECORD5
FIELD1
FIELD2
RECORD6
FIELD1
FIELD2
FIELD3
RECORD7
FIELD1
RECORD8
26
24
250
8
10
12
42
BLANK SPECIFICATION
RECORD1
RECORD2
FIELD1
RECORD3
FIELD1
RECORD11
RECORD4
FIELD1
RECORD5
FIELD1
FIELD2
RECORD6
FIELD1
FIELD2
FIELD3
RECORD7
FIELD1
RECORD8
30
28
20

FIG. 6A

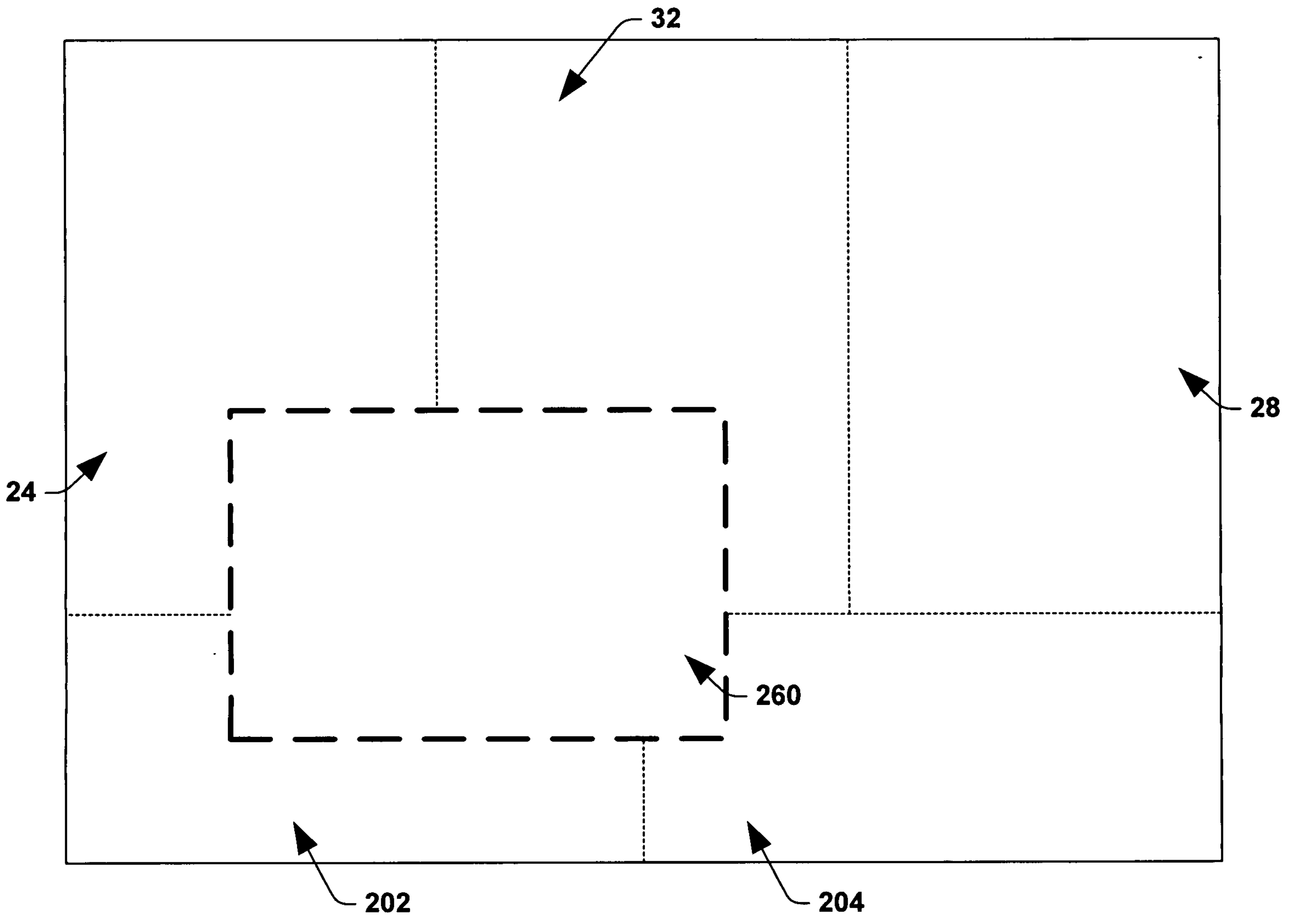
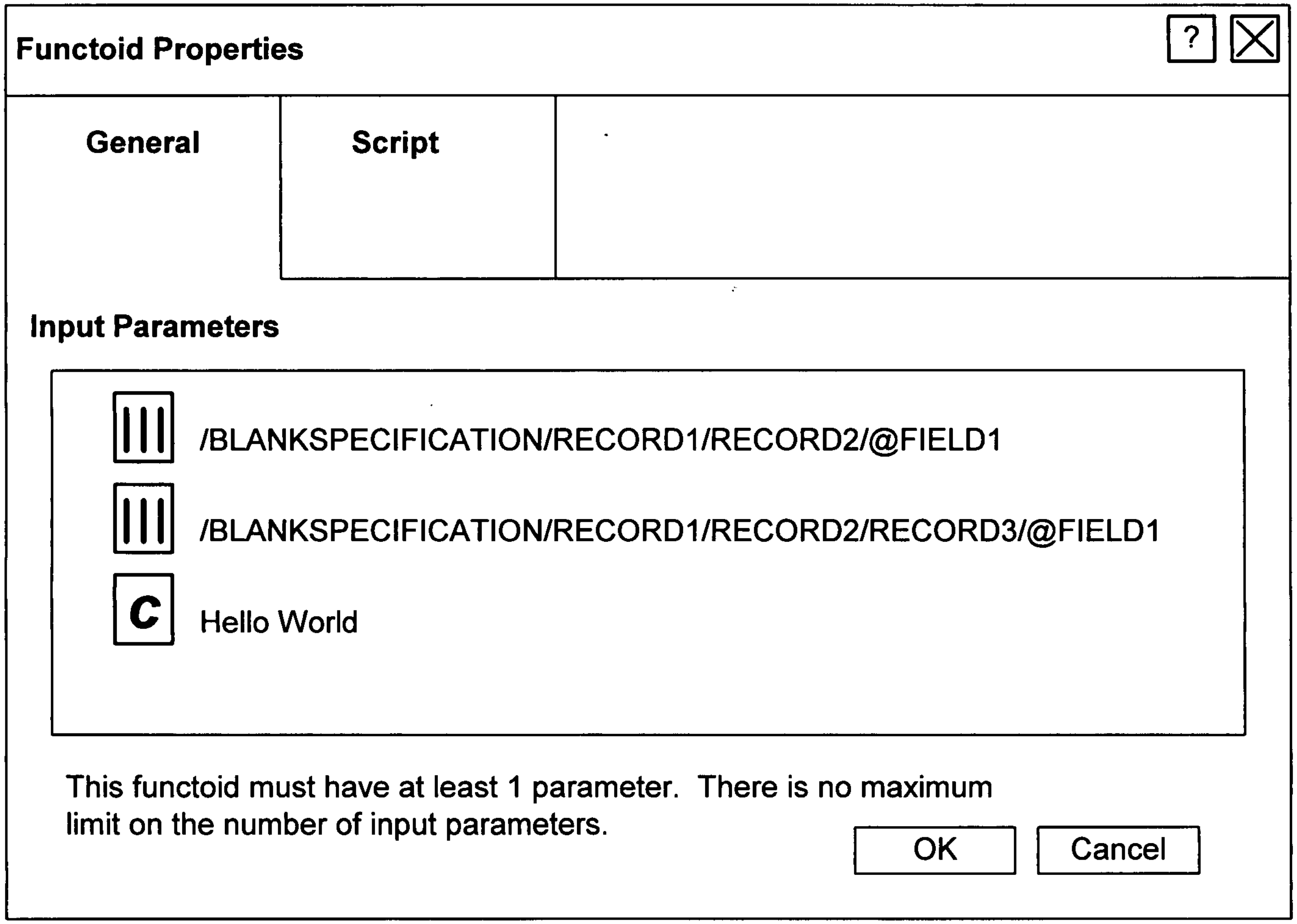


FIG. 6B



Input Parameters

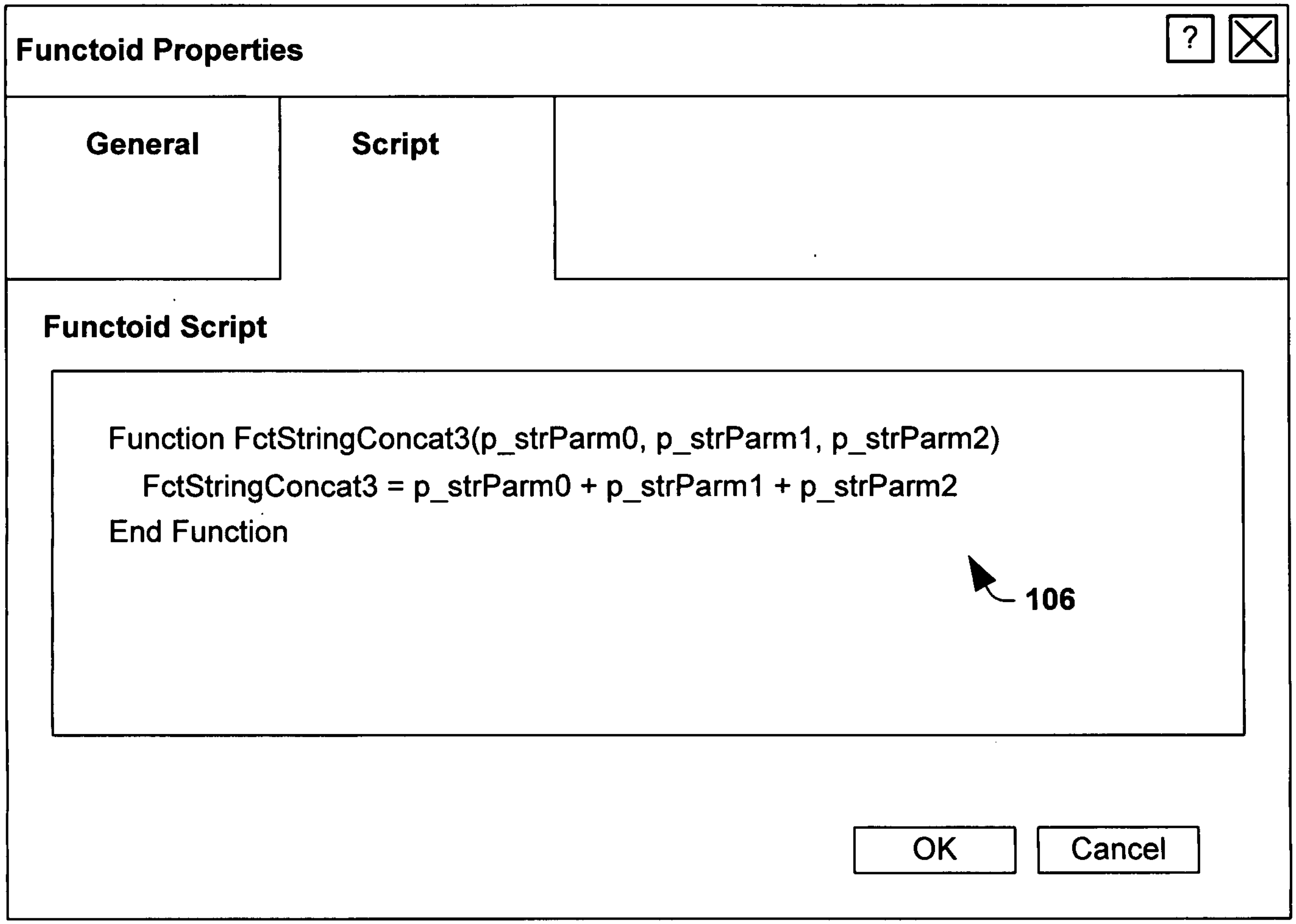
/BLANKSPECIFICATION/RECORD1/RECORD2/@FIELD1

/BLANKSPECIFICATION/RECORD1/RECORD2/RECORD3/@FIELD1

C Hello World

This functoid must have at least 1 parameter. There is no maximum limit on the number of input parameters.

FIG. 6C



```
Function FctStringConcat3(p_strParm0, p_strParm1, p_strParm2)
   FctStringConcat3 = p_strParm0 + p_strParm1 + p_strParm2
End Function
```

FIG. 6D

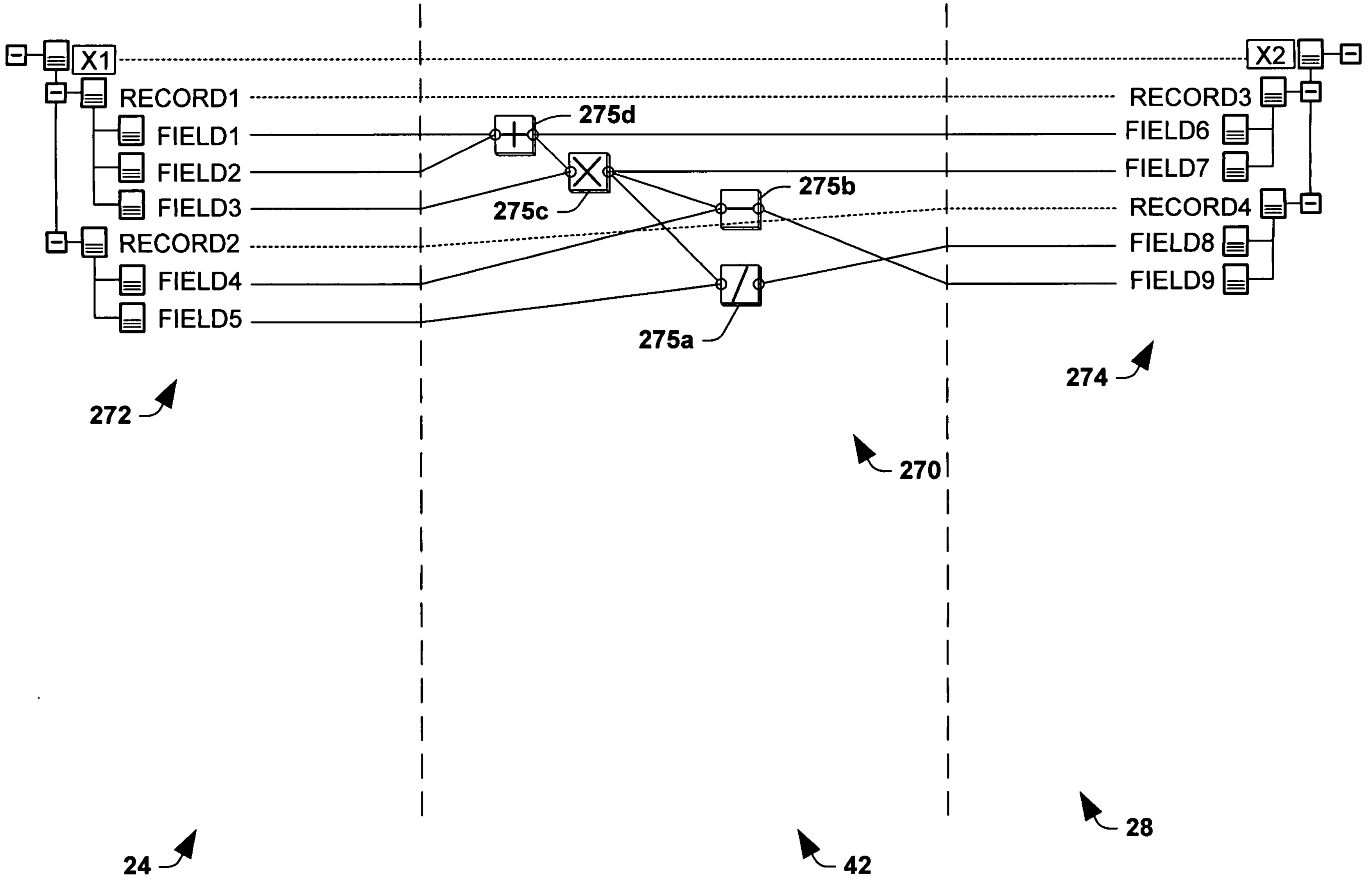

X1
RECORD1
FIELD1
FIELD2
FIELD3
RECORD2
FIELD4
FIELD5
275d
275c
275b
275a
X2
RECORD3
FIELD6
FIELD7
RECORD4
FIELD8
FIELD9
272
274
270
24
42
28
20

FIG. 7A

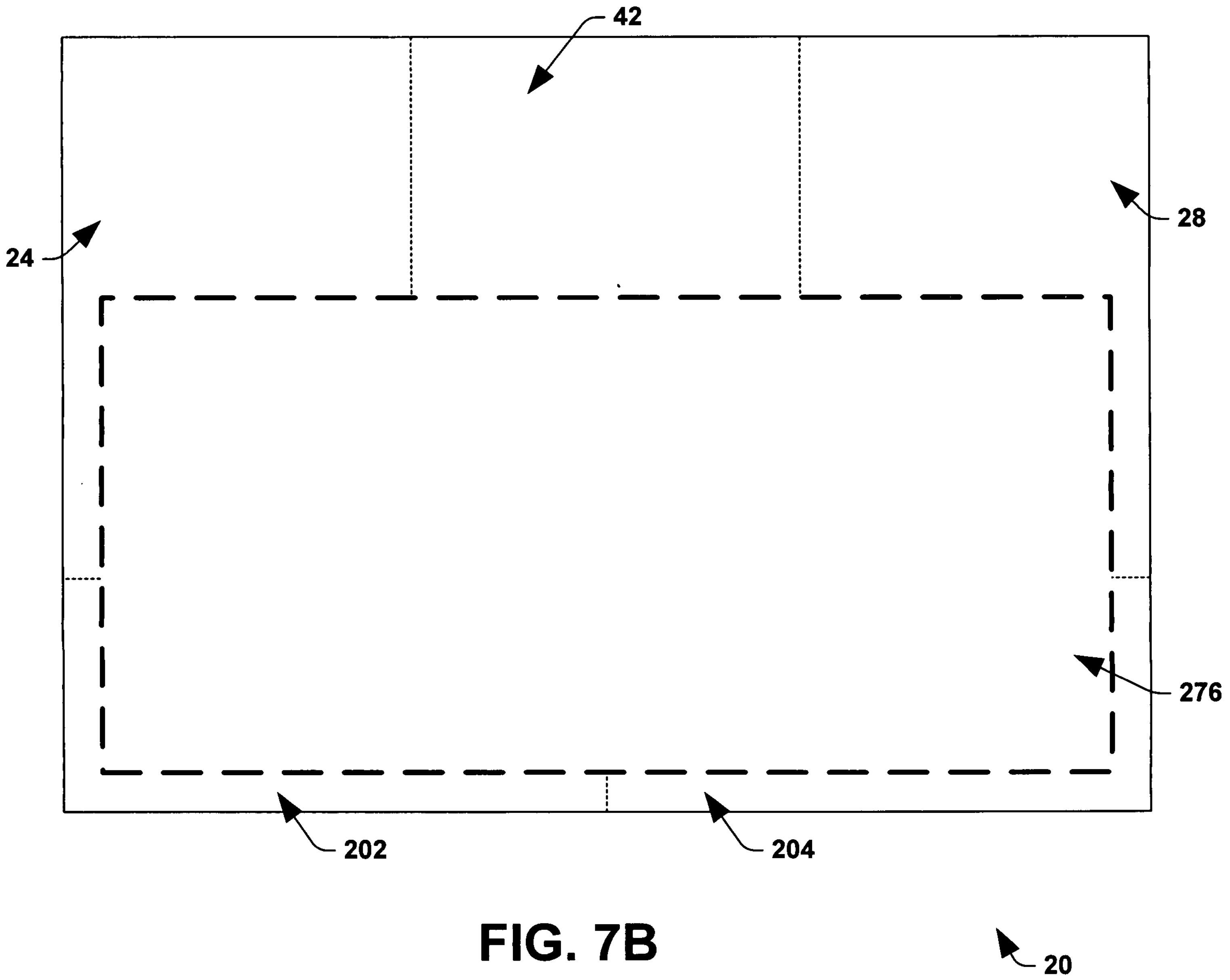
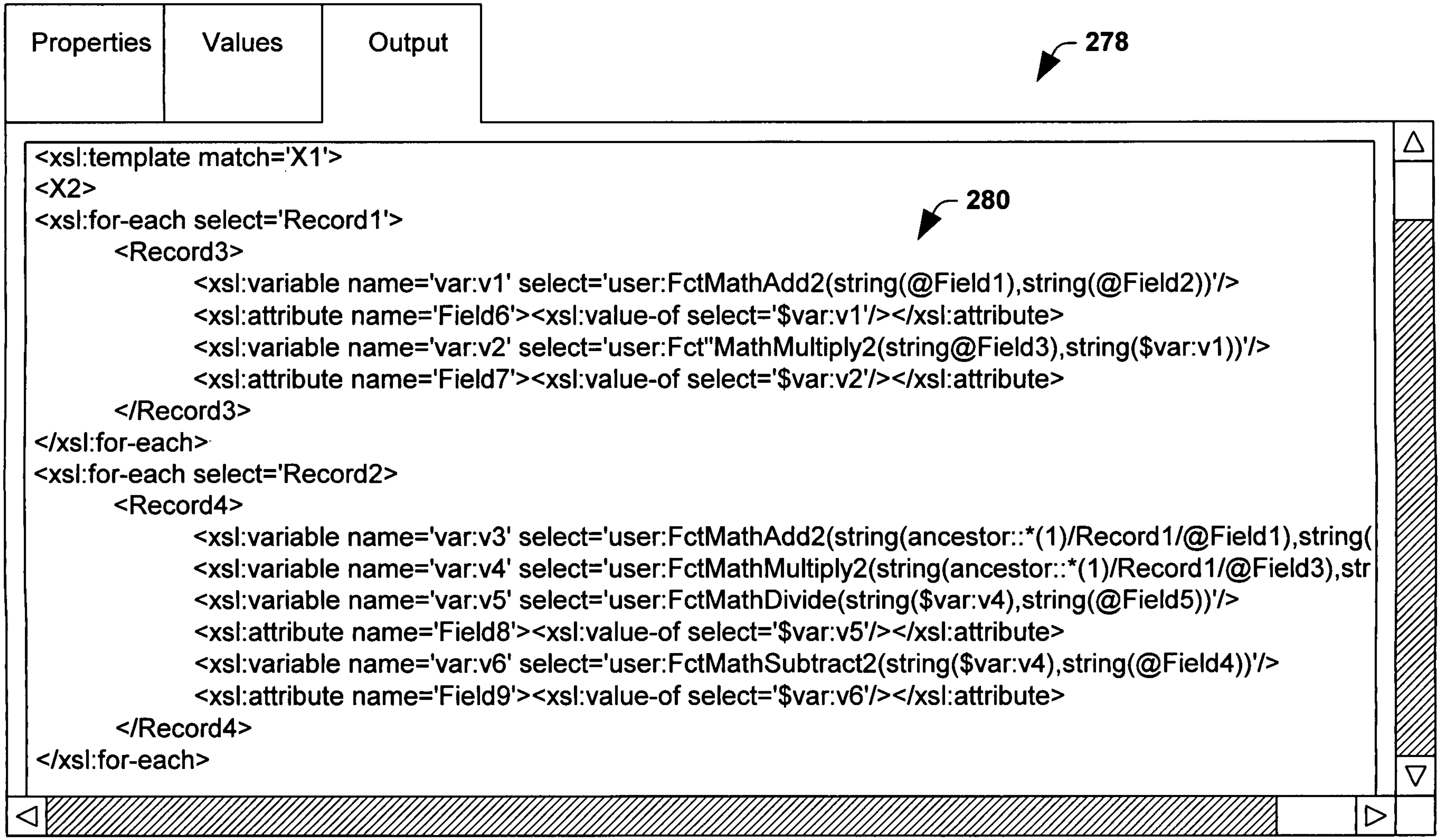


FIG. 7B



```
<xsl:template match='X1'>
<X2>
<xsl:for-each select='Record1'>
        <Record3>
                <xsl:variable name='var:v1' select='user:FctMathAdd2(string(@Field1),string(@Field2))'/>
                <xsl:attribute name='Field6'><xsl:value-of select='$var:v1'/></xsl:attribute>
                <xsl:variable name='var:v2' select='user:Fct"MathMultiply2(string@Field3),string($var:v1))'/>
                <xsl:attribute name='Field7'><xsl:value-of select='$var:v2'/></xsl:attribute>
        </Record3>
</xsl:for-each>
<xsl:for-each select='Record2>
        <Record4>
                <xsl:variable name='var:v3' select='user:FctMathAdd2(string(ancestor::*(1)/Record1/@Field1),string(
                <xsl:variable name='var:v4' select='user:FctMathMultiply2(string(ancestor::*(1)/Record1/@Field3),str
                <xsl:variable name='var:v5' select='user:FctMathDivide(string($var:v4),string(@Field5))'/>
                <xsl:attribute name='Field8'><xsl:value-of select='$var:v5'/></xsl:attribute>
                <xsl:variable name='var:v6' select='user:FctMathSubtract2(string($var:v4),string(@Field4))'/>
                <xsl:attribute name='Field9'><xsl:value-of select='$var:v6'/></xsl:attribute>
        </Record4>
</xsl:for-each>
```

FIG. 7C

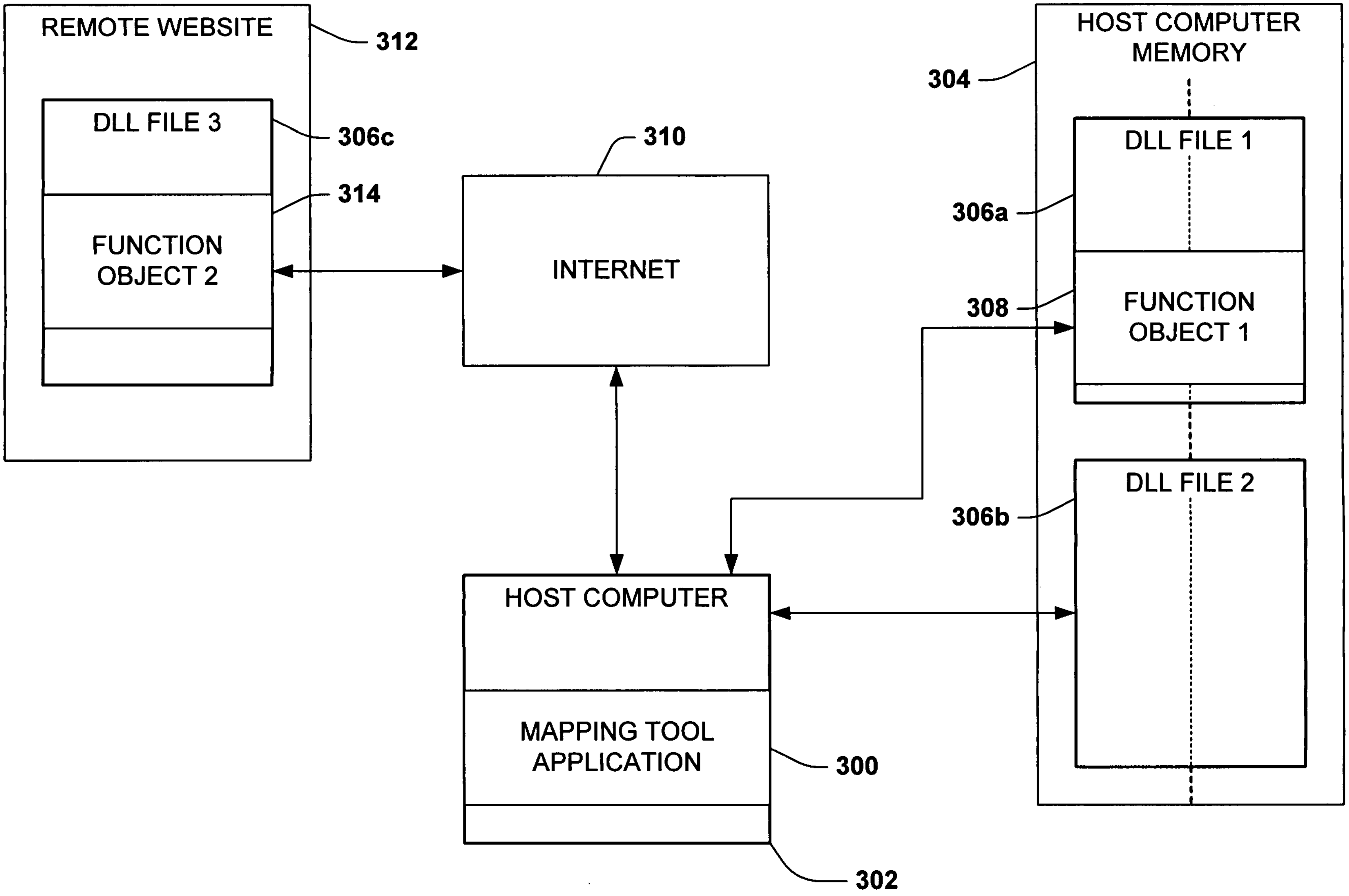

REMOTE WEBSITE
312
DLL FILE 3
306c
314
FUNCTION OBJECT 2
310
INTERNET
HOST COMPUTER MEMORY
304
DLL FILE 1
306a
308
FUNCTION OBJECT 1
DLL FILE 2
306b
HOST COMPUTER
MAPPING TOOL APPLICATION
300
302

FIG. 8

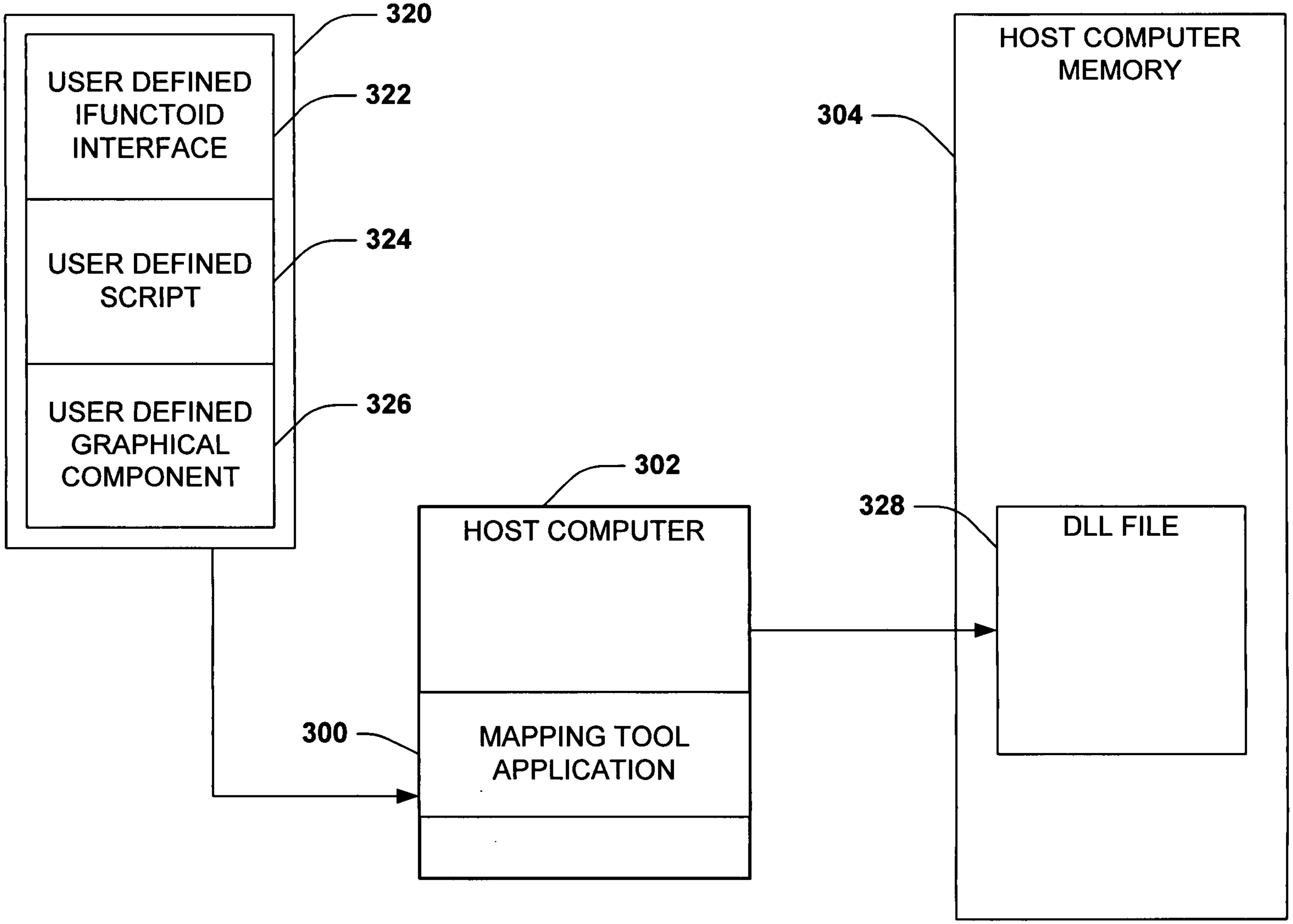

320
USER DEFINED IFUNCTOID INTERFACE
322
USER DEFINED SCRIPT
324
USER DEFINED GRAPHICAL COMPONENT
326
302
HOST COMPUTER
300
MAPPING TOOL APPLICATION
304
HOST COMPUTER MEMORY
328
DLL FILE

FIG. 9A

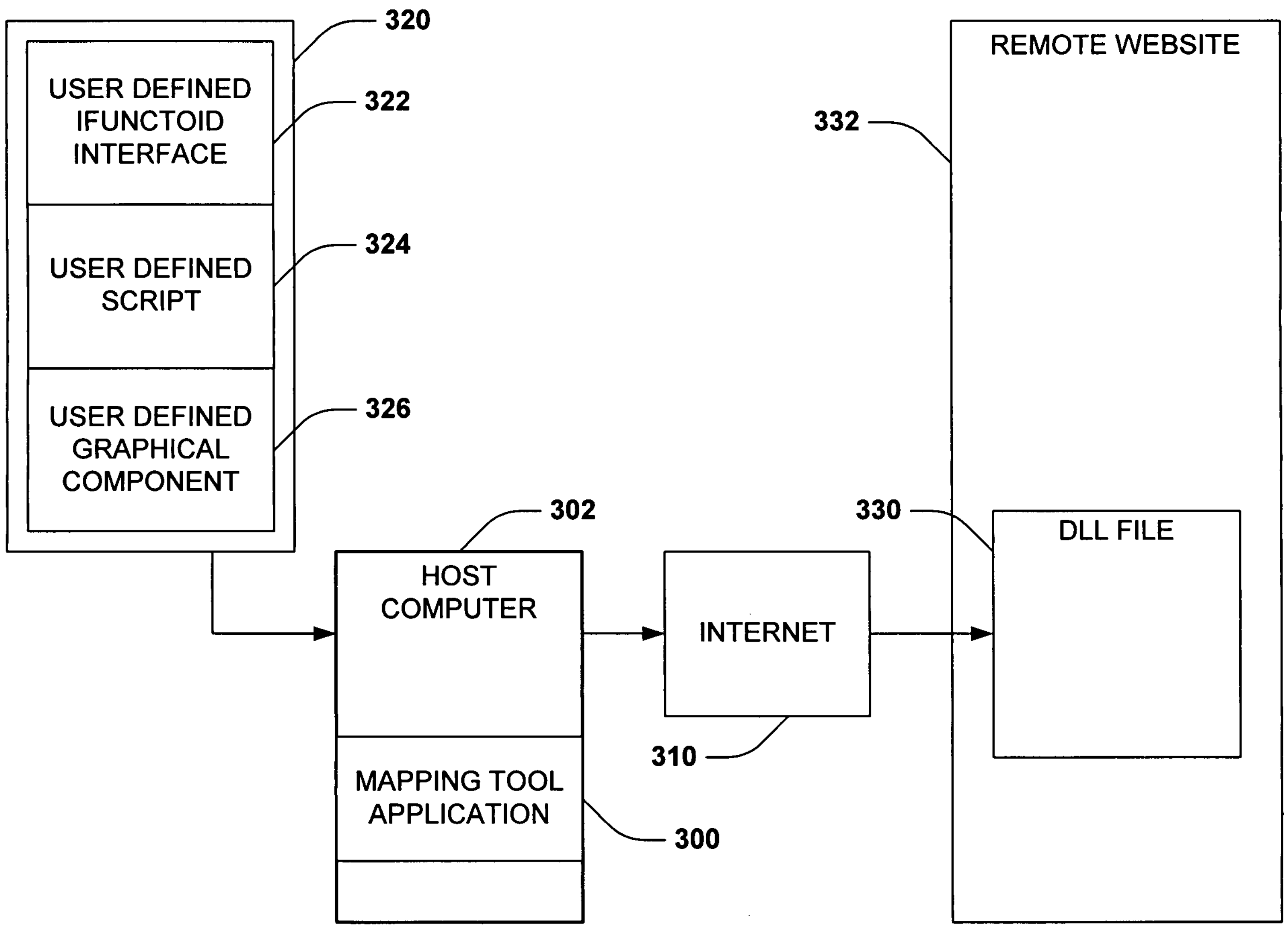

320
USER DEFINED IFUNCTOID INTERFACE
322
USER DEFINED SCRIPT
324
USER DEFINED GRAPHICAL COMPONENT
326
302
HOST COMPUTER
MAPPING TOOL APPLICATION
300
INTERNET
310
REMOTE WEBSITE
332
330
DLL FILE

FIG. 9B

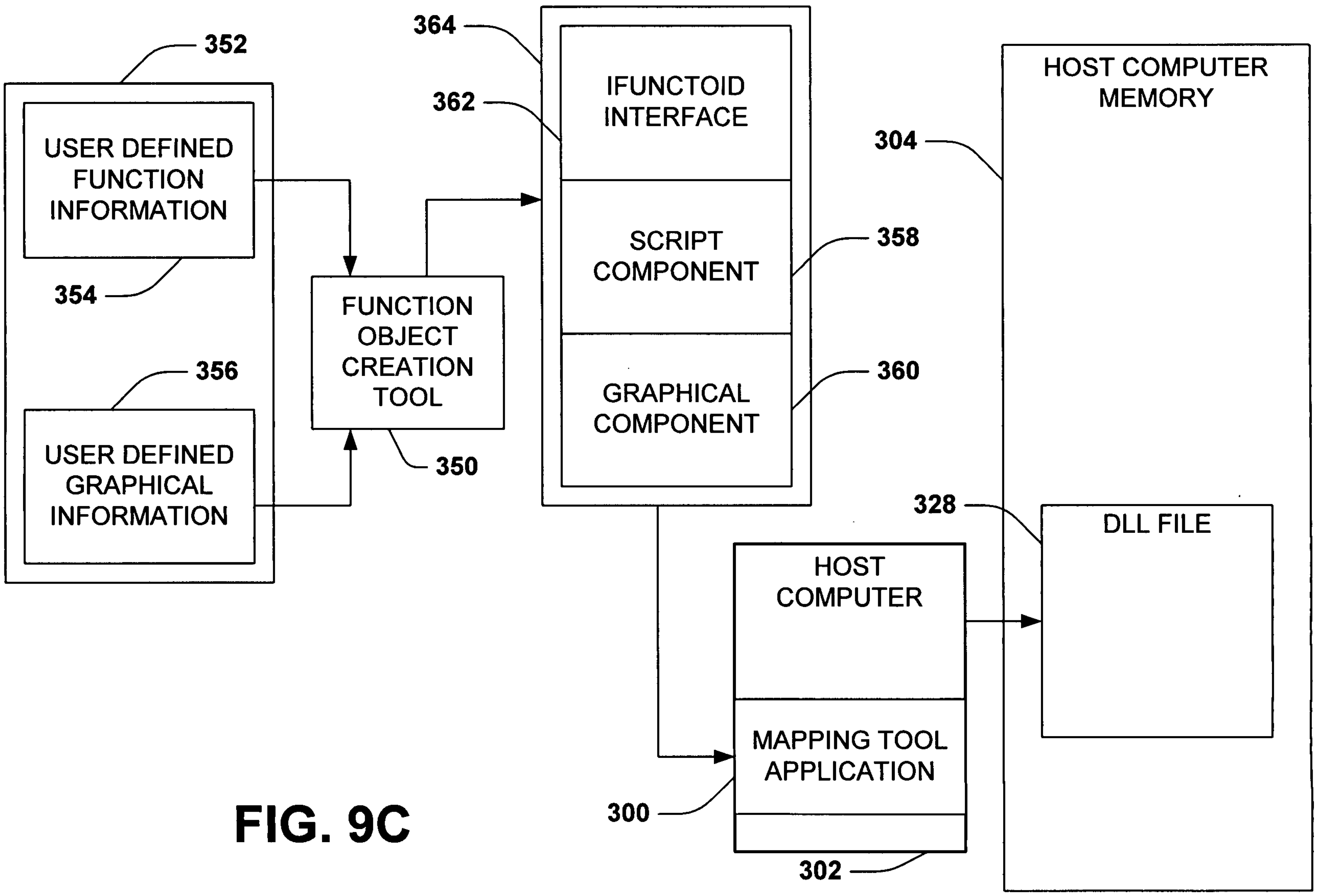

352
USER DEFINED FUNCTION INFORMATION
354
356
USER DEFINED GRAPHICAL INFORMATION
FUNCTION OBJECT CREATION TOOL
350
364
362
IFUNCTOID INTERFACE
SCRIPT COMPONENT
358
GRAPHICAL COMPONENT
360
304
HOST COMPUTER MEMORY
328
DLL FILE
HOST COMPUTER
MAPPING TOOL APPLICATION
300
302

FIG. 9C

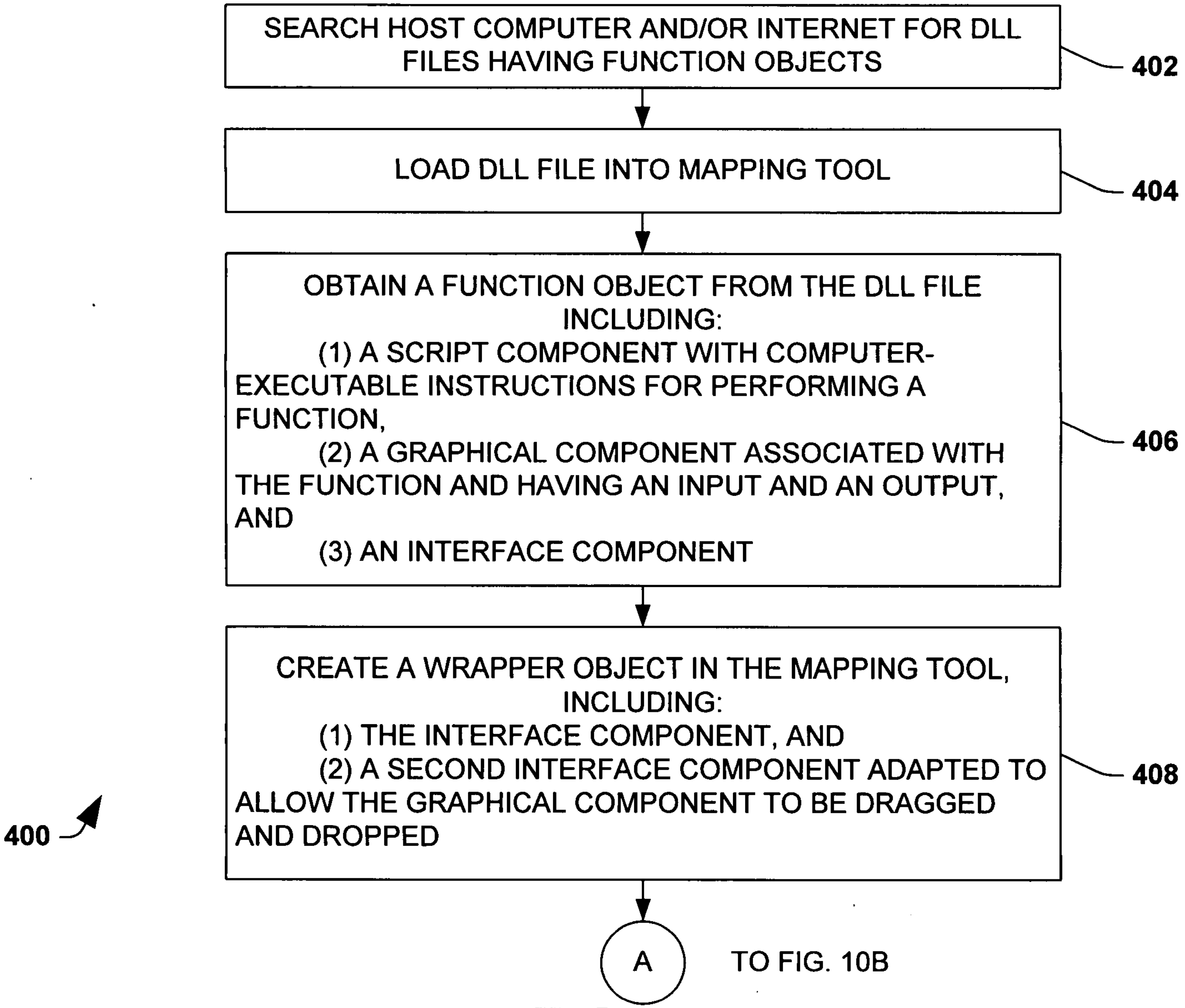

SEARCH HOST COMPUTER AND/OR INTERNET FOR DLL FILES HAVING FUNCTION OBJECTS
402
LOAD DLL FILE INTO MAPPING TOOL
404
OBTAIN A FUNCTION OBJECT FROM THE DLL FILE INCLUDING:
(1) A SCRIPT COMPONENT WITH COMPUTER-EXECUTABLE INSTRUCTIONS FOR PERFORMING A FUNCTION,
(2) A GRAPHICAL COMPONENT ASSOCIATED WITH THE FUNCTION AND HAVING AN INPUT AND AN OUTPUT, AND
(3) AN INTERFACE COMPONENT
406
CREATE A WRAPPER OBJECT IN THE MAPPING TOOL, INCLUDING:
(1) THE INTERFACE COMPONENT, AND
(2) A SECOND INTERFACE COMPONENT ADAPTED TO ALLOW THE GRAPHICAL COMPONENT TO BE DRAGGED AND DROPPED
408
400
A
TO FIG. 10B

FIG. 10A

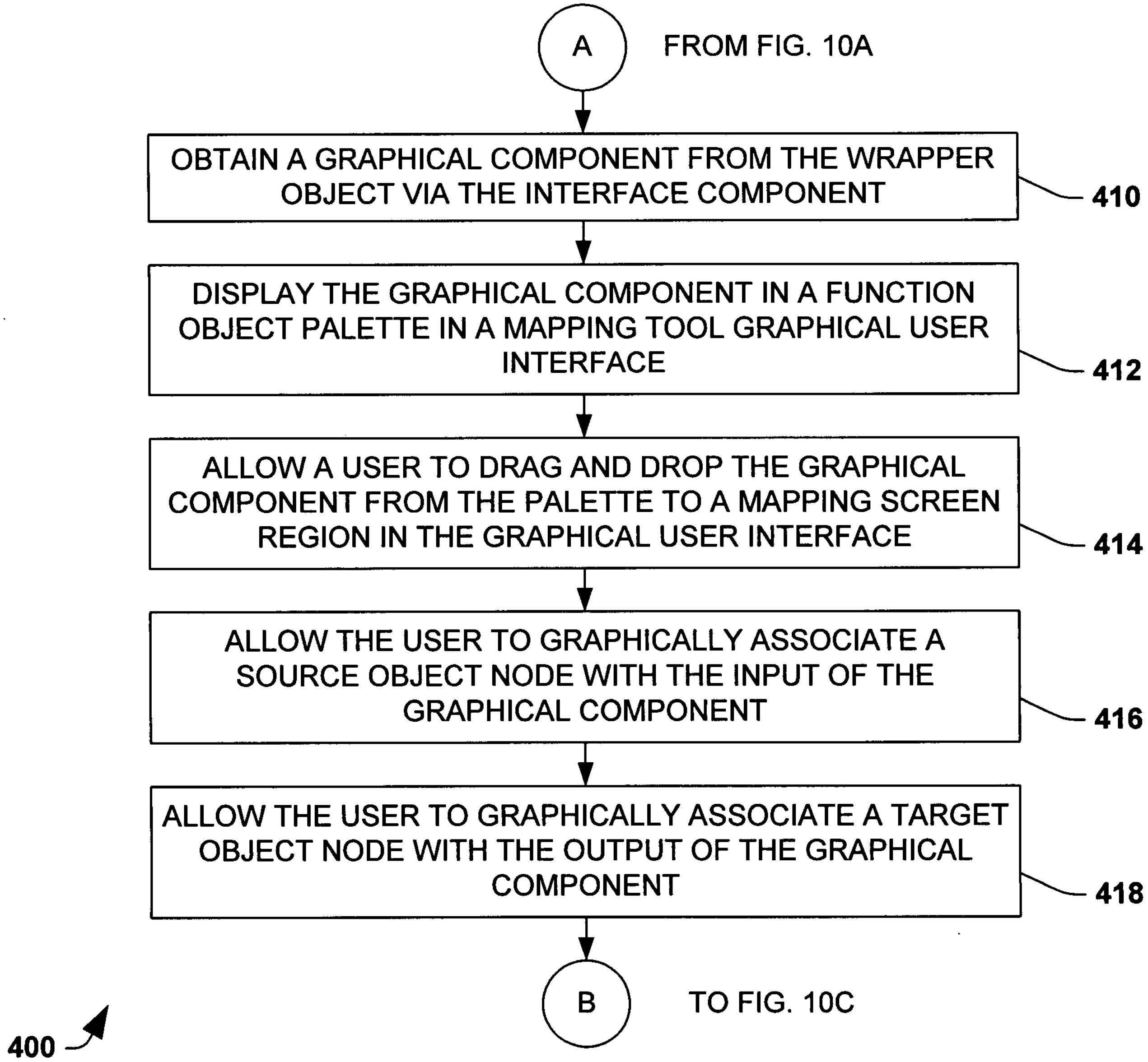

A
FROM FIG. 10A
OBTAIN A GRAPHICAL COMPONENT FROM THE WRAPPER OBJECT VIA THE INTERFACE COMPONENT
410
DISPLAY THE GRAPHICAL COMPONENT IN A FUNCTION OBJECT PALETTE IN A MAPPING TOOL GRAPHICAL USER INTERFACE
412
ALLOW A USER TO DRAG AND DROP THE GRAPHICAL COMPONENT FROM THE PALETTE TO A MAPPING SCREEN REGION IN THE GRAPHICAL USER INTERFACE
414
ALLOW THE USER TO GRAPHICALLY ASSOCIATE A SOURCE OBJECT NODE WITH THE INPUT OF THE GRAPHICAL COMPONENT
416
ALLOW THE USER TO GRAPHICALLY ASSOCIATE A TARGET OBJECT NODE WITH THE OUTPUT OF THE GRAPHICAL COMPONENT
418
B
TO FIG. 10C
400

FIG. 10B

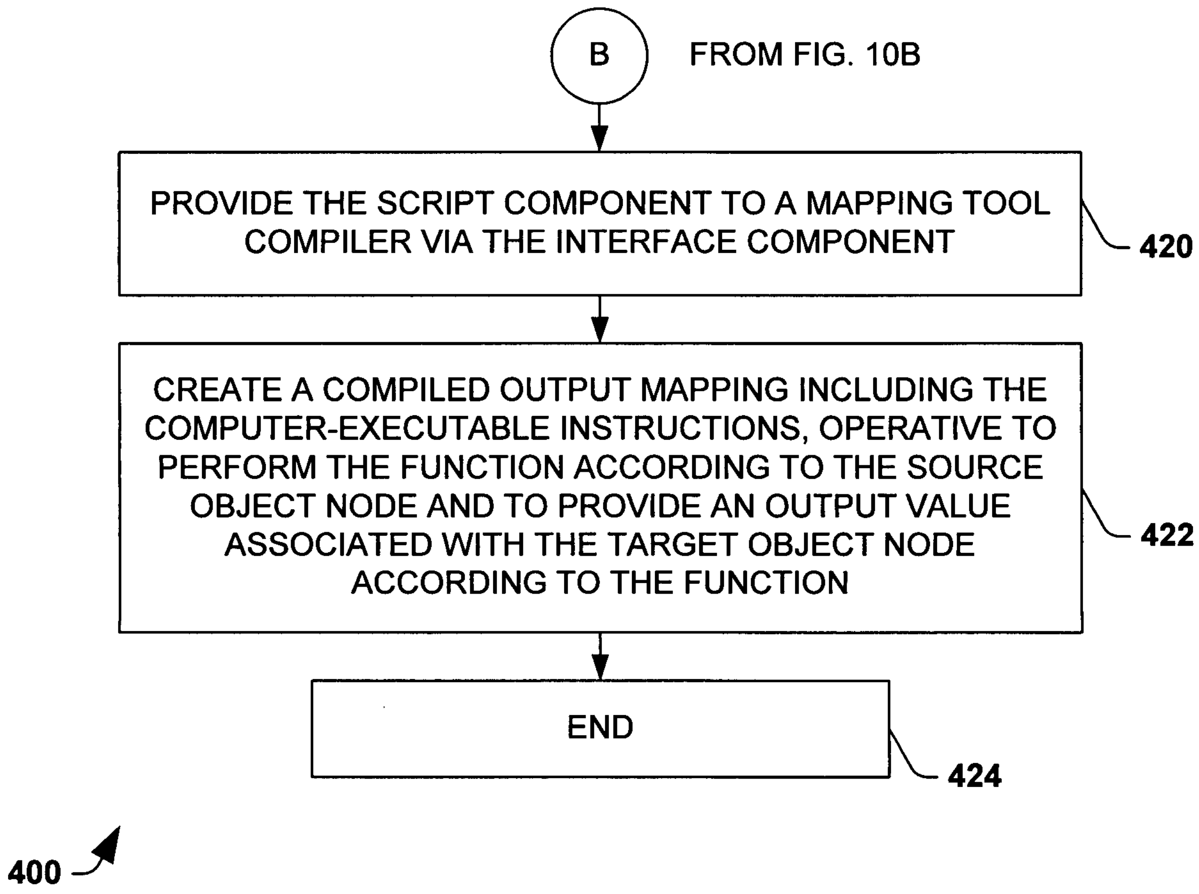

B
FROM FIG. 10B
PROVIDE THE SCRIPT COMPONENT TO A MAPPING TOOL COMPILER VIA THE INTERFACE COMPONENT
420
CREATE A COMPILED OUTPUT MAPPING INCLUDING THE COMPUTER-EXECUTABLE INSTRUCTIONS, OPERATIVE TO PERFORM THE FUNCTION ACCORDING TO THE SOURCE OBJECT NODE AND TO PROVIDE AN OUTPUT VALUE ASSOCIATED WITH THE TARGET OBJECT NODE ACCORDING TO THE FUNCTION
422
END
424
400

FIG. 10C

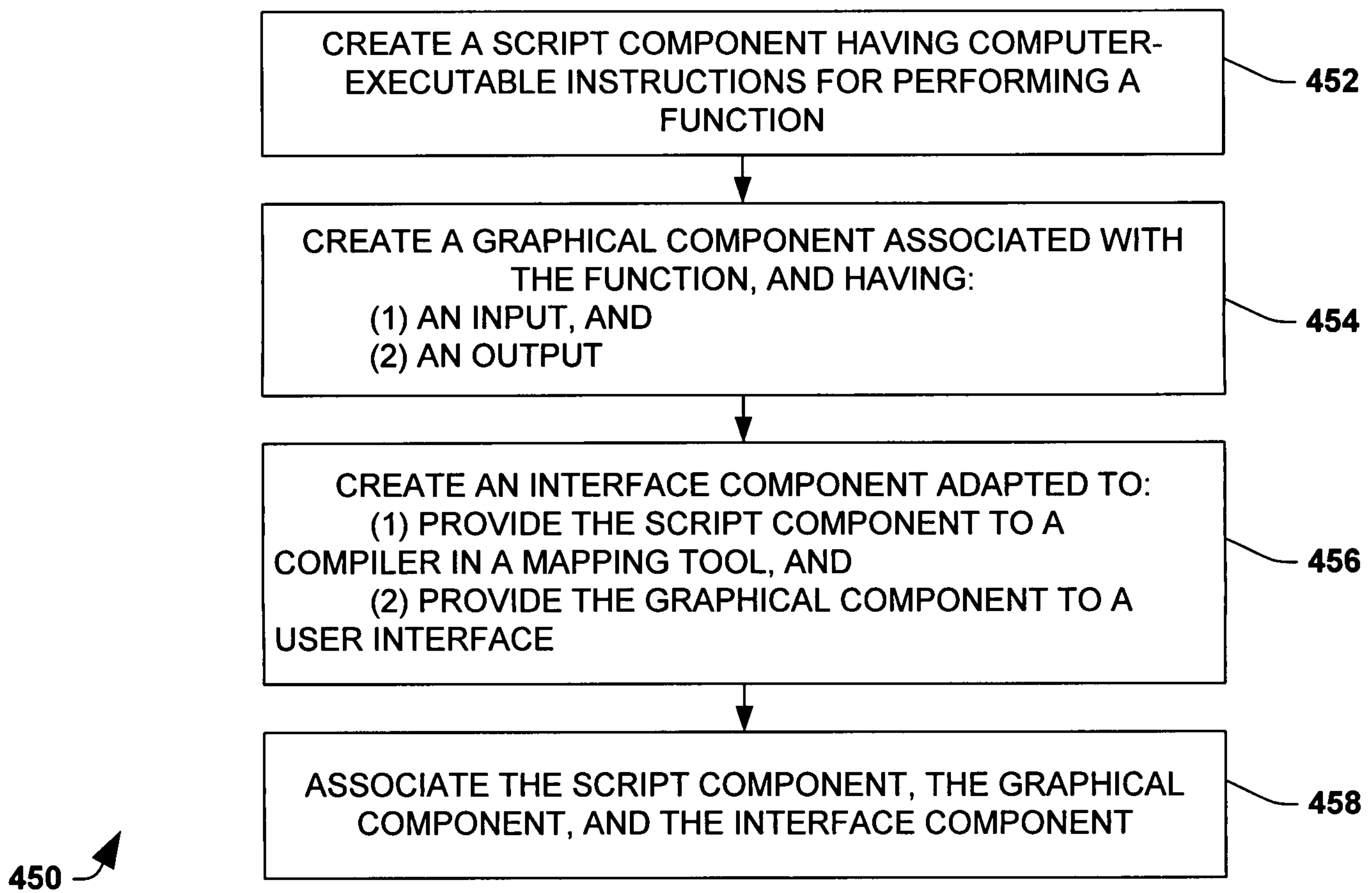

CREATE A SCRIPT COMPONENT HAVING COMPUTER-EXECUTABLE INSTRUCTIONS FOR PERFORMING A FUNCTION
452
CREATE A GRAPHICAL COMPONENT ASSOCIATED WITH THE FUNCTION, AND HAVING:
(1) AN INPUT, AND
(2) AN OUTPUT
454
CREATE AN INTERFACE COMPONENT ADAPTED TO:
(1) PROVIDE THE SCRIPT COMPONENT TO A COMPILER IN A MAPPING TOOL, AND
(2) PROVIDE THE GRAPHICAL COMPONENT TO A USER INTERFACE
456
ASSOCIATE THE SCRIPT COMPONENT, THE GRAPHICAL COMPONENT, AND THE INTERFACE COMPONENT
458
450

FIG. 11

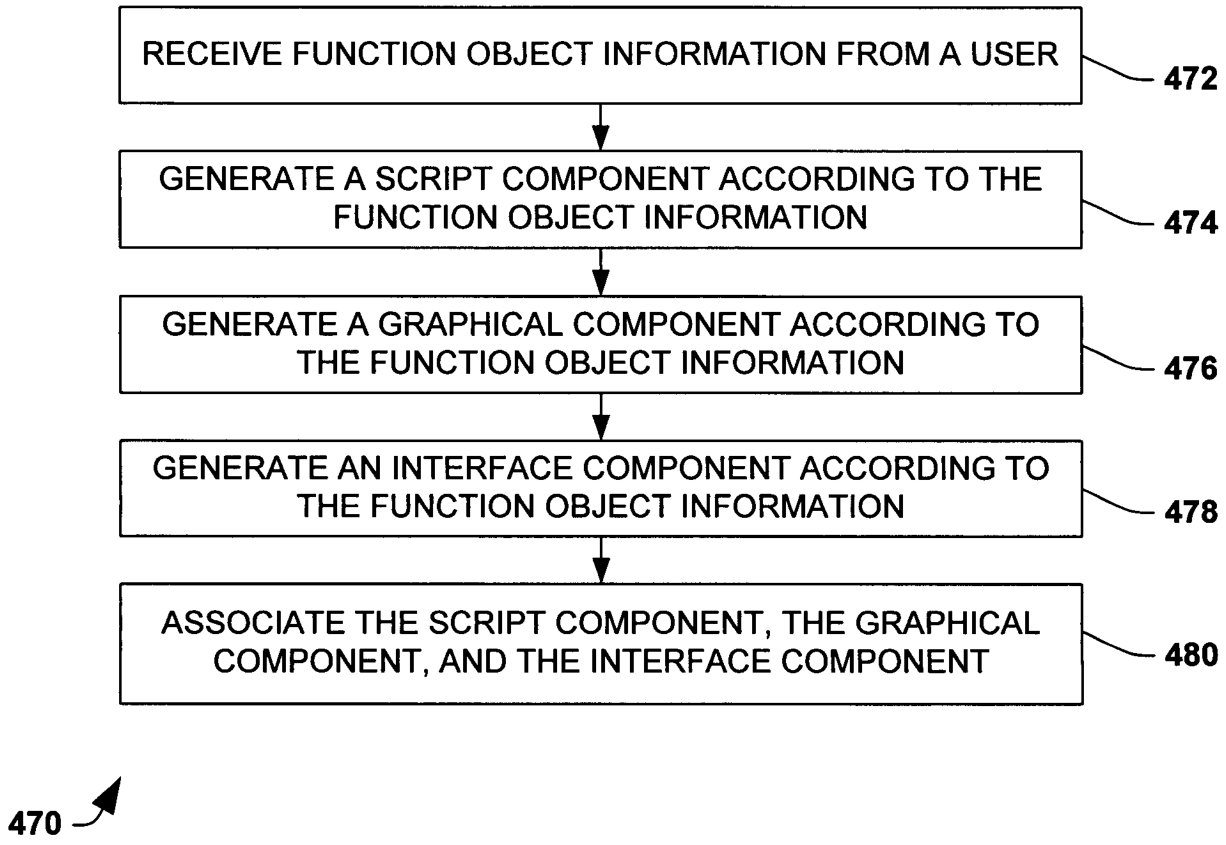

RECEIVE FUNCTION OBJECT INFORMATION FROM A USER
472
GENERATE A SCRIPT COMPONENT ACCORDING TO THE FUNCTION OBJECT INFORMATION
474
GENERATE A GRAPHICAL COMPONENT ACCORDING TO THE FUNCTION OBJECT INFORMATION
476
GENERATE AN INTERFACE COMPONENT ACCORDING TO THE FUNCTION OBJECT INFORMATION
478
ASSOCIATE THE SCRIPT COMPONENT, THE GRAPHICAL COMPONENT, AND THE INTERFACE COMPONENT
480
470

FIG. 12

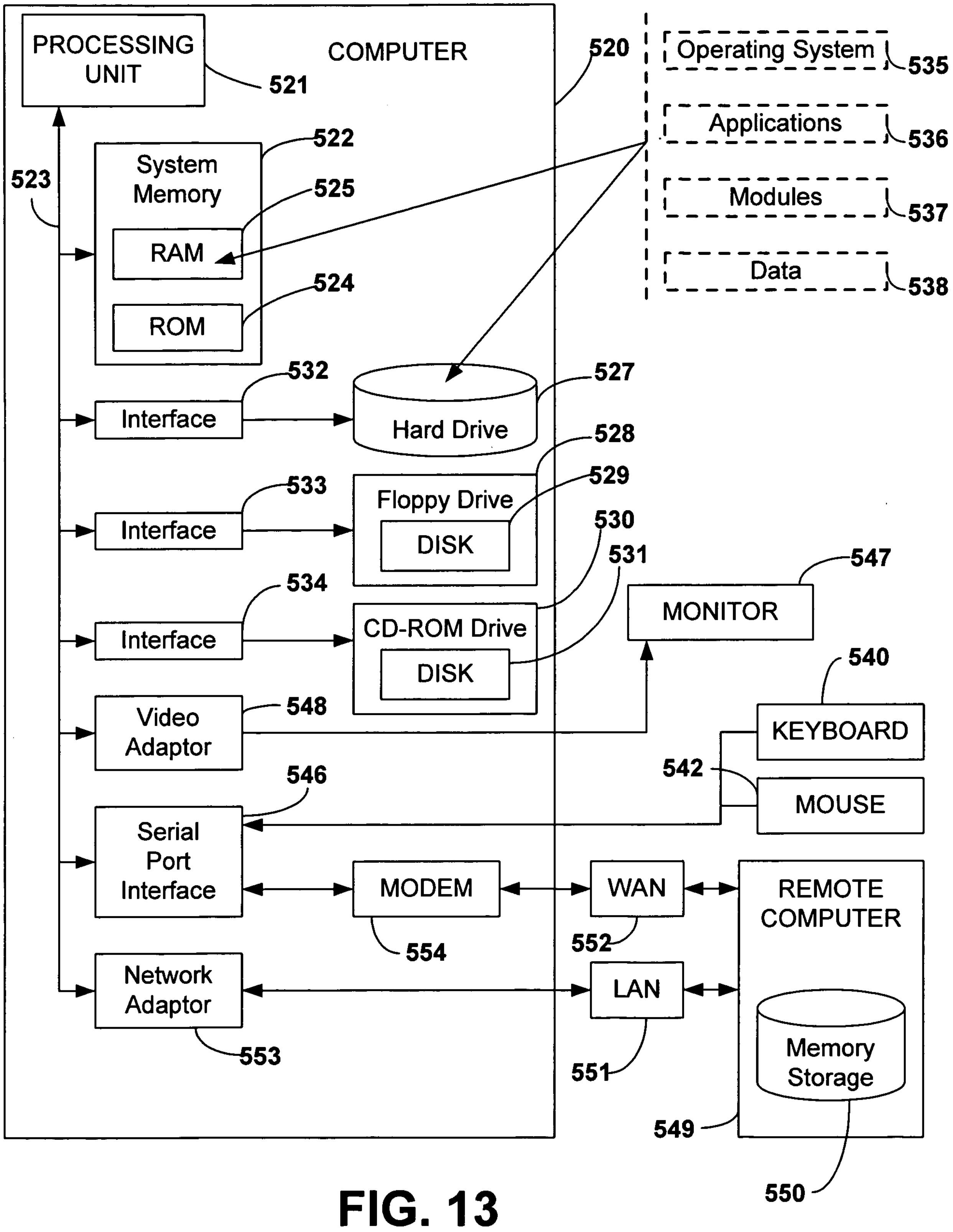

PROCESSING UNIT
521
COMPUTER
520
Operating System
535
Applications
536
Modules
537
Data
538
522
System Memory
525
RAM
523
524
ROM
532
Interface
Hard Drive
527
528
533
Floppy Drive
529
Interface
DISK
530
531
547
534
Interface
CD-ROM Drive
DISK
MONITOR
540
548
Video Adaptor
KEYBOARD
546
542
MOUSE
Serial Port Interface
MODEM
WAN
REMOTE COMPUTER
554
552
Network Adaptor
LAN
Memory Storage
553
551
549
550

FIG. 13